Remarks

Applicant respectfully requests review of this application. Claims 1-28 are currently pending. Claim 7, 16, and 25 are amended. No claims have been cancelled or added.

Thus, claims 1-28 are hereby presented for examination.

Amendments to the Claims

Claims 7, 16, and 25 have been amended to correct minor errors in these claims.

Claim Rejections under 35 U.S.C. §101

Claims 23-28 were rejected under 35 USC §101 as being directed to non-statutory subject matter.

Without any concession regarding the basis for the rejection, the Specification has been modified to remove reference to data signals embodied in a carrier wave or other propagation medium.

It is submitted that the amendment fully addresses the rejection, and the rejection should now be removed.

Claim Rejection under 35 U.S.C. §103

Devanbu, et al. and TLS Protocol

Claims 1-10, 13, and 16-26 were rejected under 35 USC §103 (a) as being unpatentable over U.S. Patent No. 6,148,401 of Devanbu, et al. (hereinafter referred to as *Devanbu*) and the Transport Layer Security (TLS) Protocol Version 1.0, RFC 2246 (hereinafter referred to as the *TLS Protocol*).

For convenience in examination, claim 1 is:

1. A method comprising:

requesting a service for a platform;

certifying the use of the service for one or more acceptable configurations of the platform; and

receiving a session key for a session of the service, the service being limited to the one or more acceptable configurations of the platform.

Claim 1 provides for requesting a service for a platform, and certifying the use of the service for one or more acceptable configurations of the platform. Claim 1 further provides for receiving a session key for a session of the service, with the session key being limited to the one or more acceptable configurations of the platform. It is respectfully submitted that the cited references do not teach or reasonably suggest these claim limitations.

Devanbu relates to a system for providing assurance that a piece of software possesses a particular property. It is submitted that the reference has no relevance to the certification of a configuration of the platform, but rather is limited to the nature of the software that is being certified.

In particular, *Devanbu* describes a determination whether a set of instructions possess a particular property. For example, the summary of *Devanbu* indicate that "a system and method provide assurance to a host that a set of subject instructions adapted to be executed on a host processor possess a property." (*Devanbu*, col. 4, lines 27-30) This is indicating that the set of subject instructions (the software in question) possesses a certain property, without any regard to whether platform has any particular configuration. As the process is described:

> In one embodiment, a verification processor executes a version of a set of verification instructions to determine if the set of subject instructions possess the property. If the set of subject instructions possess the property, then the verification processor cryptographically signs the set of instructions to produce signature information, and in one embodiment of the present invention, distributes the set of instructions with the signature information. In one embodiment, information pertaining to the property verified by the provider can be derived by a host from the set of subject instructions and the signature data. In another embodiment, the provider cryptographically signs property data identifying the property of the set of subject instructions verified by the provider.

(*Devanbu*, col. 4, lines 31-43) In the described process, a verification processor will execute verification instructions to determine if the set of subject instructions possesses a particular property. If this is true, then the verification processor will provide a cryptographic signature for the instructions.

The remainder of the summary then describes how the signature to verify the presence of the property in the subject instructions is used to evaluate the integrity and authenticity of the instructions:

> When a host receives the set of subject instructions and the signature, the host can use the signature to determine the integrity and the authenticity of the subject set of instructions, as well as the identity of the property verified by the provider. If the host cannot certify the set of subject instructions and the property data using the signature information, then the host does not execute the software. If the host can certify the set of subject instructions and the property data, then the host may execute the software.

(*Devanbu*, col. 4, lines 44-52) Thus, if the host can verify the instructions and the property based on the signature, the instructions are executed, and will not execute the instructions otherwise.

The process may be further seen in, for example, Figures 5 and 6 of *Devanbu*. As the process is illustrated in Figure 5, the software is received 341 and there is determination whether the software possesses a particular property 342. If not, the process returns. If the software does posses the property, then the software is signed and a signature is provided in a certificate 343, and the software and certificate is sent to the host 344. Figure 6 then provides that the software and certificate is received 351. If the signature is valid 352, then the software possesses the particular property in question and can be executed 353. Otherwise, the software is not executed 354.

Figures 1, 2, and 3 then illustrate the system in which this process may be implemented. Figure 1 provides a physically secure coprocessor system for the verification and certification process, which includes a processor 102 that operates with a memory, a private cryptographic key 104, and certification instructions 108. Figure 2 provides an alternative arrangement utilizing an ASIC 201 rather than a processor. The system is then illustrated in Figure 3, with the physically secure coprocessor 403 operating in conjunction with a software provider 402. The claims are consistent with this process. Claim 1 of *Devanbu*, for example, describes "method for providing assurance to a host that a set of subject instructions possesses a particular property", including determining if the set of subject instructions possesses the particular property at a certifier, and, if the subject set of instructions is determined to possess the particular

property, then signing the set of subject instructions at the certifier and distributing to the host the set of subject instructions and a certificate that includes the signature.

As to the identity of the "particular property" that a set of subject instructions might possess, the reference indicates:

> An example of a property of a piece of software is the identity of the author of the software. Another example of a property is the identity of the compiler used to generate the piece of software. In certain applications, it is important to provide assurance to the host that a piece of software cannot alter the contents of a file stored on a disk drive of the computer on which the software is executed. This is another example of a property of a piece of software.

(*Devanbu*, col. 1, lines 26-33) Thus, the examples provided are the author of the software, the compiler used to generate the software, and the ability of the software to alter a certain file. In connection with this, the reference also describes the parties who might provide the software, and describes what might be the host executing the software. (*Devanbu*, col. 1, lines 34-54) The reference then claims certain embodiments of the particular property in the software, which may be the certifier of the subject instructions (Claim 2, although this may not make any sense because the certifier is the party doing the certification), the identity of the compiler to generate the binary version of the instructions (Claim 3), and the manufacturer of the instructions (Claim 4). In each case, the property is a property of the software itself or the identity of an agent that has created, processed, or manufactured the software. There is no teach or suggestion of any connection with acceptable configurations of the platform for which a service is requested.

The *TLS Protocol* is intended to provide privacy and data integrity between two communicating applications. (See, *TLS Protocol*, Introduction, p. 3) To accomplish this goal, the *TLS Protocol* includes a record protocol and a handshake protocol, with the record protocol being used to encapsulate higher level protocols and the handshake protocol being used to allow a server and client to authenticate each other and negotiate an encryption algorithm and cryptographic keys before an application protocol transmits or receives any data. (*TLS Protocol*, Introduction, p. 4) While the TLS protocol provides tools for secure communications, this does not have any direct relation to the elements of the claims in the present application, and the *TLS Protocol* does not contain the claim elements that have been shown above to be missing from *Devanbu*. There is no provision in the *TLS Protocol* that relates to certifying the use of a service for one or more acceptable configurations of the platform, or to a session key for a session of the service that is limited to the one or more acceptable configurations of the platform.

It is submitted that the arguments presented above are also applicable to the other independent claim provided in the present application, claim 8 (method including validating a service key, with validating comprising receipt of assurance that the service is used only for one or more acceptable configurations for the platform), claim 13 (claim for a client device, which is to provide assurance to a service provider that a service is limited to one or more acceptable configurations for the platform), claim 16 (claim for a system, including a client device to certify that a service will be utilized only in one or more acceptable configurations of a platform of the client device), and claims 23 and 26 (claims for readable medium). Thus, claims 8, 13, 16, 23, and 26 are also allowable. The

remaining rejected claims, while having other differences with the cited references, are allowable as being dependent on the allowable base claims.

Claim Rejection under 35 U.S.C. §103

Devanbu, et al., TLS Protocol, and Todd, et al.

Claims 11, 14, and 27 were rejected under 35 USC §103 (a) as being unpatentable over *Devanbu* and the *TLS Protocol* as applied to claims 8, 13, and 26, and further in view of U.S. Patent No. 5,867,714 of Todd, et al. (hereinafter referred to as *Todd*).

The rejected claims, while having other differences with the cited references, are allowable as being dependent on the allowable base claims.

Devanbu and the *TLS Protocol* have been addressed above. It is further submitted that the *Todd* reference, while cited for other purposes, does not contain the claim limitations that, as shown above, are not taught or suggested by *Devanbu* and the *TLS Protocol*. *Todd* relates to a system for distributing configuration dependent software revisions to a computer system. As described in the Abstract:

> There is disclosed a system for detecting and avoiding faults stemming from conflicts in hardware and/or software configurations in a computer system. The system comprises communications circuitry that, from time to time, automatically transmits at least part of the current hardware and software configuration data of the computer system to a remote data source capable of identifying inherent conflicts in the hardware and software configuration. The remote data source then transmits to the computer system software revisions that are capable of resolving the inherent conflicts. After the communications circuitry receives the software revisions, processing circuitry in the computer system modifies the current software configuration according to the received software revisions.

(*Todd*, Abstract) Thus, the system involves issues regarding hardware and software configurations, but in regard to ensuring the correct software is delivered to a system. This is accomplished by transmitting configuration data to a data source capable of identifying inherent conflicts in the hardware and software configurations, and the data source transmitting software revisions that are capable of resolving the conflicts.

Thus, *Todd* involves a different issue, which is the delivery of software that addresses configuration conflicts in a system. *Todd* is not relevant to certifying the use of a service for one or more acceptable configurations of the platform, or to a session key for a session of the service that is limited to the one or more acceptable configurations of the platform.

Claim Rejection under 35 U.S.C. §103

Devanbu, et al., TLS Protocol, and Klayh, et al.

Claims 12 and 15 were rejected under 35 USC §103 (a) as being unpatentable over *Devanbu* and the *TLS Protocol*, as applied to claims 8 and 13, and further in view of International Patent Publication WO/2000/038089 of Klayh, et al. (hereinafter referred to as *Klayh*).

The rejected claims, while having other differences with the cited references, are allowable as being dependent on the allowable base claims.

Devanbu and the *TLS Protocol* have been addressed above. It is further submitted that the *Klayh* reference, while cited for other purposes, does not contain the claim limitations that, as shown above, are not taught or suggested by *Devanbu* and the *TLS Protocol*. *Klayh* describes an amusement and premiums network, involving a system for controlling a medium of distribution and redemption of loyalty points and coupons.

Klayh is not relevant to certifying the use of a service for one or more acceptable configurations of the platform, or to a session key for a session of the service that is limited to the one or more acceptable configurations of the platform.

Claim Rejection under 35 U.S.C. §103

Devanbu, et al., TLS Protocol, Todd, et al., and Klayh, et al.

Claims 28 was rejected under 35 USC §103 (a) as being unpatentable over *Devanbu*, the *TLS Protocol*, and *Todd* as applied to claim 27, and further in view of *Klayh*.

The rejected claim, while having other differences with the cited references, is allowable as being dependent on the allowable base claims. All of the references have been addressed above.

Conclusion

Applicant respectfully submits that the rejections have been overcome by the amendment and remark, and that the claims as amended are now in condition for allowance. Accordingly, Applicant respectfully requests the rejections be withdrawn and the claims as amended be allowed.

Invitation for a Telephone Interview

The Examiner is requested to call the undersigned at (503) 439-8778 if there remains any issue with allowance of the case.

Request for an Extension of Time

The Applicant respectfully petitions for extension of time to respond to the outstanding Office Action pursuant to 37 C.F.R. § 1.136(a) should one be needed. Please change the fee under 37 C.F.R. § 1.17 for such extension to our Deposit Account No. 02-2666.

Charge our Deposit Account

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 4/27/2007 /Mark C. Van Ness/

Mark C. Van Ness
Reg. No. 39,865

12400 Wilshire Boulevard
7th Floor
Los Angeles, California 90025-1026
(503) 439-8778